

Notice of Allowability

Application No.

10/701,388

Examiner

Stephen Elmore

Applicant(s)

TENE ET AL.

Art Unit

2185

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address--

All claims being allowable, PROSECUTION ON THE MERITS IS (OR REMAINS) CLOSED in this application. If not included herewith (or previously mailed), a Notice of Allowance (PTOL-85) or other appropriate communication will be mailed in due course. **THIS NOTICE OF ALLOWABILITY IS NOT A GRANT OF PATENT RIGHTS.** This application is subject to withdrawal from issue at the initiative of the Office or upon petition by the applicant. See 37 CFR 1.313 and MPEP 1308.

1. ☒ This communication is responsive to the application filed November 4, 2003.
2. ☒ The allowed claim(s) is/are 1-28 and 32-81 (renumbered as 1-78).
3. ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
 - a) ☐ All b) ☐ Some* c) ☐ None of the:
 1. ☐ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. _____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this national stage application from the International Bureau (PCT Rule 17.2(a)).

* Certified copies not received: _____.

Applicant has THREE MONTHS FROM THE "MAILING DATE" of this communication to file a reply complying with the requirements noted below. Failure to timely comply will result in ABANDONMENT of this application.
THIS THREE-MONTH PERIOD IS NOT EXTENDABLE.

4. ☐ A SUBSTITUTE OATH OR DECLARATION must be submitted. Note the attached EXAMINER'S AMENDMENT or NOTICE OF INFORMAL PATENT APPLICATION (PTO-152) which gives reason(s) why the oath or declaration is deficient.
5. ☐ CORRECTED DRAWINGS (as "replacement sheets") must be submitted.
 - (a) ☐ including changes required by the Notice of Draftsperson's Patent Drawing Review (PTO-948) attached
 - 1) ☐ hereto or 2) ☐ to Paper No./Mail Date _____.
 - (b) ☐ including changes required by the attached Examiner's Amendment / Comment or in the Office action of Paper No./Mail Date _____.Identifying indicia such as the application number (see 37 CFR 1.84(c)) should be written on the drawings in the front (not the back) of each sheet. Replacement sheet(s) should be labeled as such in the header according to 37 CFR 1.121(d).
6. ☐ DEPOSIT OF and/or INFORMATION about the deposit of BIOLOGICAL MATERIAL must be submitted. Note the attached Examiner's comment regarding REQUIREMENT FOR THE DEPOSIT OF BIOLOGICAL MATERIAL.

Attachment(s)

1. ☒ Notice of References Cited (PTO-892) ✓
2. ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3. ☒ Information Disclosure Statements (PTO-1449 or PTO/SB/08),
Paper No./Mail Date 2/6/2004
4. ☐ Examiner's Comment Regarding Requirement for Deposit
of Biological Material

5. ☐ Notice of Informal Patent Application (PTO-152)
6. ☒ Interview Summary (PTO-413) ✓
Paper No./Mail Date this paper.
7. ☒ Examiner's Amendment/~~Comment~~ ✓
8. ☒ Examiner's Statement of Reasons for Allowance ✓
9. ☒ Other "Attachment A"

6/22/2006
SC En
STEPHEN C. ELMORE
PRIMARY EXAMINER

INTERVIEW SUMMARY, EXAMINER'S AMENDMENT and REASONS FOR ALLOWANCE

INTERVIEW SUMMARY

1. A examiner-initiated telephonic interview was conducted on June 20, 2006 with Applicant's Representative, Mr. Lee Van Pelt, where the following issues were discussed:

a. A restriction of the claims in the case was considered proper by the Examiner based on the following analysis:

Claims 1-28 were considered properly placed in Classification 711/154, Electrical Computers and Digital Processing Systems: Memory/Control Technique;

Claims 29-31 were deemed to be more properly placed in Classification 717/151, Data Processing: Software Development, Installation, and Management/Optimization.

Mr. Van Pelt agreed to the proposed restriction, without traverse, and gave the Examiner permission to cancel claims 29-31 in an Examiner's Amendment.

b. The Examiner argued that in respect to the remaining claims, these claims appear allowable over the discovered prior art of record, except for one problem pertaining to independent claim 27, which was that this claim recited the limitation "a computer readable medium," which limitation under the present computer-implemented inventions guidelines appears to read over non-statutory subject matter because the Specification, at page 5, lines 3-5, suggests that the scope of "computer readable medium" encompasses non-statutory subject matter because the computer readable medium includes a "computer network wherein program instructions are sent over optical or electronic communications links," which subject matter includes non-tangible implementations of computer readable medium. Claim 27's deficiency could be cured by limiting the scope of "computer readable medium" to "tangible" implementations. Mr. Van Pelt agreed to permit an Examiner's Amendment to Claim 27 to amend the claim to recite "a tangible computer readable medium" in order to further prosecution of the claims.

c. In view of the above discussion, Mr. Van Pelt requested the Examiner to consider a proposed amendment to the claims to add new dependent claims to independent claims 27 and 28, these being new dependent claims 32-81, which claims would cover the allowable subject matter

Art Unit: 2185

corresponding to the dependent method claims of independent claim 1. The Examiner agreed to consider such a proposed amendment.

2. The attached document labeled "Preliminary Amendment A," dated June 21, 2006, is the proposed amendment previously discussed and has been labeled "Attachment A" by the Examiner.

3. A second telephonic interview was made on June 22, 2006, after review of the proposed amendment, to Applicant's Representative Ms. Diana Fu, Reg. No. 52, 924. The interview was required because the proposed amendment failed to properly apply the "tangible" limitation to independent claim 27 and to proposed dependent claims 32-56 in accordance with the discussion with Mr. Van Pelt on June 20, 2006.

In the second interview, Ms. Diana Fu authorized the Examiner to make changes to the affected proposed claims to modify the "computer readable medium" in claim 27 into "a tangible computer readable medium" and to remove all other remaining references to "tangible" in the proposed claims. This amendment overcomes the 101 issue with claim 27 and its dependent claims 32-56.

4. Therefore, after performing a final search of the prior art and after considering the attached proposed amendment and the second interview, the following Examiner's Amendment is provided that has been approved by the Applicant's Representative Ms. Diana Fu in the above Examiner-Initiated Interview.

5. Finally, Deposit Account No. 50-0685 (AZULP002) has been approved for payment of additional claim fees and/or extension of time fees, according to page 2 of the FACSIMILE COVER SHEET ("Attachment A") dated June 21, 2006, signed by Applicant's Representative, Diana Y. Fu, Reg. No. 52,924.

Art Unit: 2185

EXAMINER'S AMENDMENT

6. An examiner's amendment to the record appears below. Should the changes and/or additions be unacceptable to applicant, an amendment may be filed as provided by 37 CFR 1.312. To ensure consideration of such an amendment, it MUST be submitted no later than the payment of the issue fee.

7. Authorization for this examiner's amendment was given in a telephone interview with Applicant's Representative, Lee Van Pelt, Reg. No. 38,353, on June 20, 2006 and in a second telephone interview with Applicant's Representative, Diana Y. Fu, Reg. No. 52,924, on June 22, 2006.

The application has been amended as follows:

IN THE CLAIMS

Claim 27, line 1, replace "a computer readable memory" with --a tangible computer readable memory--;

Claims 29-31 are canceled.

Add dependent Claims 32-81 as provided in the attached listing of all the claims which begins on the next page.

IN THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Original) A method for managing an object in memory, comprising:
 - assigning the object to an assigned frame wherein the object can be released when the assigned frame is released;
 - detecting an attempt to place a reference to the object in an older frame, the older frame being older than the assigned frame; and
 - reassigning the object to a reassignment frame that is at least as old as the older frame.
2. (Original) A method for managing an object in memory as recited in Claim 1, wherein the reassignment frame is the older frame.
3. (Original) A method for managing an object in memory as recited in Claim 1, wherein assigning the object to the assigned frame comprises associating a frame identifier with the object.
4. (Original) A method for managing an object in memory as recited in Claim 1, wherein assigning the object to the assigned frame comprises associating a frame identifier with a reference of the object.
5. (Original) A method for managing an object in memory as recited in Claim 1, wherein assigning the object to the assigned frame comprises associating a frame identifier with the object and detecting an attempt to place a reference to the object in an older frame is performed using the frame identifier.
6. (Original) A method for managing an object in memory as recited in Claim 1, wherein detecting an attempt to place a reference to the object in an older frame comprises comparing a first frame identifier associated with the object with a second frame identifier

Art Unit: 2185

associated with the older frame.

7. (Original) A method for managing an object in memory as recited in Claim 1, wherein detecting an attempt to place a reference to the object in an older frame comprises comparing a first address associated with the object with a second address associated with the older frame.

8. (Original) A method for managing an object in memory as recited in Claim 1, wherein detecting an attempt to place a reference to the object in an older frame comprises determining whether the object is in stack memory or heap memory.

9. (Original) A method for managing an object in memory as recited in Claim 1, wherein detecting an attempt to place a reference to the object in an older frame comprises determining whether the object is in stack memory or heap memory by examining a distinguishing bit or a distinguishing set of bits.

10. (Original) A method for managing an object in memory as recited in Claim 1, wherein detecting an attempt to place a reference to the object in an older frame comprises determining whether the object is in stack memory or heap memory; and heap memory is uniquely identified by a heap frame identifier.

11. (Original) A method for managing an object in memory as recited in Claim 1, wherein reassigning the object to a reassignment frame that is at least as old as the older frame comprises recursively detecting whether the object references any younger object.

12. (Original) A method for managing an object in memory as recited in Claim 1, wherein reassigning the object to a reassignment frame that is at least as old as the older frame comprises recursively detecting whether the object references any younger object and reassigning any referenced younger object to the reassignment frame.

13. (Original) A method for managing an object in memory as recited in Claim 1, wherein reassigning the object to a reassignment frame comprises resetting a first frame identifier associated with the object to be the same as a second frame identifier associated with the reassignment frame.

Art Unit: 2185

14. (Original) A method for managing an object in memory as recited in Claim 1, wherein reassigning the object to a reassignment frame comprises moving the object into the reassignment frame.
15. (Original) A method for managing an object in memory as recited in Claim 1, wherein reassigning the object to a reassignment frame comprises moving the object into the reassignment frame and storing overflow in an overflow area associated with the reassignment frame.
16. (Original) A method for managing an object in memory as recited A Claim 1, wherein reassigning the object to a reassignment frame comprises expanding the reassignment frame and moving the object into the reassignment frame.
17. (Original) A method for managing an object in memory as recited in Claim 1, wherein reassigning the object to a reassignment frame comprises moving the object into the reassignment frame and updating a reference to the object.
18. (Original) A method for managing an object in memory as recited in Claim 1, wherein reassigning the object to a reassignment frame comprises moving the object into the reassignment frame and updating all references to the object.
19. (Original) A method for managing an object in memory as recited in Claim 1, wherein reassigning the object to a reassignment frame is assisted by a display.
20. (Original) A method for managing an object in memory as recited in Claim 1, wherein reassigning the object includes tracing reassigned space.
21. (Original) A method for managing an object in memory as recited in Claim 1, further comprises modifying an allocation site of the object.
22. (Original) A method for managing an object in memory as recited in Claim 1, further comprises modifying a frame creation site.
23. (Original) A method for managing an object in memory as recited in Claim 1, further comprises learning reassignment information.
24. (Original) A method for managing an object in memory as recited in Claim 1, further

Art Unit: 2185

comprising performing thread-local garbage collection.

25. (Original) A method for managing an object in memory as recited in Claim 1, further comprising storing call path information associated with an allocation site of the object.

26. (Original) A method for managing an object in memory as recited in Claim 1, wherein detecting an attempt to place a reference to the object in an older frame is performed with hardware assist.

27. (Currently Amended) A computer program product for managing an object in memory, the computer program product being embodied in a tangible computer readable medium and comprising computer instructions for:

assigning the object to an assigned frame wherein the object can be released when the assigned frame is released;

detecting an attempt to place a reference to the object in an older frame, the older frame being older than the assigned frame;

reassigning the object to a reassignment frame that is at least as old as the older frame.

28. (Original) A system for managing an object, comprising:

a processor configured to:

assign the object to an assigned frame wherein the object can be released when the assigned frame is released;

detect an attempt to place a reference to the object in an older frame, the older frame being older than the assigned frame;

reassign the object to a reassignment frame that is at least as old as the older frame; and a memory coupled to the processor, configured to provide the processor with instructions.

29. (Cancelled)

30. (Cancelled)

31. (Cancelled)

32. (New) A computer program product for managing an object in memory as recited in Claim 27, wherein the reassignment frame is the older frame.

Art Unit: 2185

33. (New) A computer program product as recited in Claim 27, wherein assigning the object to the assigned frame comprises associating a frame identifier with the object.
34. (New) A computer program product as recited in Claim 27, wherein assigning the object to the assigned frame comprises associating a frame identifier with a reference of the object.
35. (New) A computer program product as recited in Claim 27, wherein assigning the object to the assigned frame comprises associating a frame identifier with the object and detecting an attempt to place a reference to the object in an older frame is performed using the frame identifier.
36. (New) A computer program product as recited in Claim 27, wherein detecting an attempt to place a reference to the object in an older frame comprises comparing a first frame identifier associated with the object with a second frame identifier associated with the older frame.
37. (New) A computer program product as recited in Claim 27, wherein detecting an attempt to place a reference to the object in an older frame comprises comparing a first address associated with the object with a second address associated with the older frame.
38. (New) A computer program product as recited in Claim 27, wherein detecting an attempt to place a reference to the object in an older frame comprises determining whether the object is in stack memory or heap memory.
39. (New) A computer program product as recited in Claim 27, wherein detecting an attempt to place a reference to the object in an older frame comprises determining whether the object is in stack memory or heap memory by examining a distinguishing bit or a distinguishing set of bits.
40. (New) A computer program product as recited in Claim 27, wherein detecting an attempt to place a reference to the object in an older frame comprises determining whether the object is in stack memory or heap memory; and heap memory is uniquely identified by a heap frame identifier.
41. (New) A computer program product as recited in Claim 27, wherein reassigning the object to a reassignment frame that is at least as old as the older frame comprises recursively detecting whether the object references any younger object.
42. (New) A computer program product as recited in Claim 27, wherein reassigning the object to a reassignment frame that is at least as old as the older frame comprises recursively detecting whether

Art Unit: 2185

the object references any younger object and reassigning any referenced younger object to the reassignment frame.

43. (New) A computer program product as recited in Claim 27, wherein reassigning the object to a reassignment frame comprises resetting a first frame identifier associated with the object to be the same as a second frame identifier associated with the reassignment frame.

44. (New) A computer program product as recited in Claim 27, wherein reassigning the object to a reassignment frame comprises moving the object into the reassignment frame.

45. (New) A computer program product as recited in Claim 27, wherein reassigning the object to a reassignment frame comprises moving the object into the reassignment frame and storing overflow in an overflow area associated with the reassignment frame.

46. (New) A computer program product as recited in Claim 27, wherein reassigning the object to a reassignment frame comprises expanding the reassignment frame and moving the object into the reassignment frame.

47. (New) A computer program product as recited in Claim 27, wherein reassigning the object to a reassignment frame comprises moving the object into the reassignment frame and updating a reference to the object.

48. (New) A computer program product as recited in Claim 27, wherein reassigning the object to a reassignment frame comprises moving the object into the reassignment frame and updating all references to the object.

49. (New) A computer program product as recited in Claim 27, wherein reassigning the object to a reassignment frame is assisted by a display.

50. (New) A computer program product as recited in Claim 27, wherein reassigning the object includes tracing reassigned space.

51. (New) A computer program product as recited in Claim 27, the computer program product further comprising instructions for modifying an allocation site of the object.

52. (New) A computer program product as recited in Claim 27, the computer program product further comprising instructions for modifying a frame creation site.

Art Unit: 2185

53. (New) A computer program product as recited in Claim 27, the computer program product further comprising instructions for learning reassignment information.

54. (New) A computer program product as recited in Claim 27, the computer program product further comprising instructions for performing thread-local garbage collection.

55. (New) A computer program product as recited in Claim 27, the computer program product further comprising instructions for storing call path information associated with an allocation site of the object.

56. (New) A computer program product as recited in Claim 27, wherein detecting an attempt to place a reference to the object in an older frame is performed with hardware assist.

57. (New) A system as recited in Claim 28, wherein the reassignment frame is the older frame.

58. (New) A system as recited in Claim 28, wherein to assign the object to the assigned frame comprises to associate a frame identifier with the object.

59. (New) A system as recited in Claim 28, wherein to assign the object to the assigned frame comprises to associate a frame identifier with a reference of the object.

60. (New) A system as recited in Claim 28, wherein to assign the object to the assigned frame comprises to associate a frame identifier with the object and to detect an attempt to place a reference to the object in an older frame is performed using the frame identifier.

61. (New) A system as recited in Claim 28, wherein to detect an attempt to place a reference to the object in an older frame comprises to compare a first frame identifier associated with the object with a second frame identifier associated with the older frame.

62. (New) A system as recited in Claim 28, wherein to detect an attempt to place a reference to the object in an older frame comprises to compare a first address associated with the object with a second address associated with the older frame.

63. (New) A system as recited in Claim 28, wherein to detect an attempt to place a reference to the object in an older frame comprises to determine whether the object is in stack memory or heap memory.

64. (New) A system as recited in Claim 28, wherein to detect an attempt to place a reference

Art Unit: 2185

to the object in an older frame comprises to determine whether the object is in stack memory or heap memory by examining a distinguishing bit or a distinguishing set of bits.

65. (New) A system as recited in Claim 28, wherein to detect an attempt to place a reference to the object in an older frame comprises to determine whether the object is in stack memory or heap memory; and heap memory is uniquely identified by a heap frame identifier.

66. (New) A system as recited in Claim 28, wherein to reassign the object to a reassignment frame that is at least as old as the older frame comprises to detect recursively whether the object references any younger object.

67. (New) A system as recited in Claim 28, wherein to reassign the object to a reassignment frame that is at least as old as the older frame comprises to detect recursively whether the object references any younger object and reassigning any referenced younger object to the reassignment frame.

68. (New) A system as recited in Claim 28, wherein to reassign the object to a reassignment frame comprises to reset a first frame identifier associated with the object to be the same as a second frame identifier associated with the reassignment frame.

69. (New) A system as recited in Claim 28, wherein to reassign the object to a reassignment frame comprises to move the object into the reassignment frame.

70. (New) A system as recited in Claim 28, wherein to reassign the object to a reassignment frame comprises to move the object into the reassignment frame and storing overflow in an overflow area associated with the reassignment frame.

71. (New) A system as recited in Claim 28, wherein to reassign the object to a reassignment frame comprises to expand the reassignment frame and to move the object into the reassignment frame.

72. (New) A system as recited in Claim 28, wherein to reassign the object to a reassignment frame comprises to move the object into the reassignment frame and to update a reference to the object.

73. (New) A system as recited in Claim 28, wherein to reassign the object to a reassignment frame comprises to move the object into the reassignment frame and to update all references to the object.

Art Unit: 2185

74. (New) A system as recited in Claim 28, wherein to reassign the object to a reassignment frame is assisted by a display.

75. (New) A system as recited in Claim 28, wherein to reassign the object includes to trace reassigned space.

76. (New) A system as recited in Claim 28, wherein the processor is further configured to modify an allocation site of the object.

77. (New) A system as recited in Claim 28, wherein the processor is further configured to modify a frame creation site.

78. (New) A system as recited in Claim 28, wherein the processor is further configured to learn reassignment information.

79. (New) A system as recited in Claim 28, wherein the processor is further configured to perform thread-local garbage collection.

80. (New) A system as recited in Claim 28, wherein the processor is further configured to store call path information associated with an allocation site of the object.

81. (New) A system as recited in Claim 28, wherein to detect an attempt to place a reference to the object in an older frame is performed with hardware assist.

REASONS FOR ALLOWANCE

8. The following is an examiner's statement of reasons for allowance:

In independent claims 1, 27 and 28 directed towards a method, a computer program product, and a system, the following features taken in combination with the remaining limitations of the independent claims are not found in and or are not obvious in view of the prior art of record,

Claim 1

"detecting an attempt to place a reference to the object in an older frame, the older frame being older than the assigned frame; and reassigning the object to a reassignment frame that is at least as old as the older frame";

Claim 27,

"detecting an attempt to place a reference to the object in an older frame, the older frame being older than the assigned frame; reassigning the object to a reassignment frame that is at least as old as the older frame";

Claim 28,

"detect an attempt to place a reference to the object in an older frame, the older frame being older than the assigned frame; reassign the object to a reassignment frame that is at least as old as the older frame; and a memory coupled to the processor, configured to provide the processor with instructions".

9. Any comments considered necessary by applicant must be submitted no later than the payment of the issue fee and, to avoid processing delays, should preferably accompany the issue fee. Such submissions should be clearly labeled "Comments on Statement of Reasons for Allowance."


Any inquiry concerning this communication or earlier communications from the examiner should be directed to Stephen Elmore whose telephone number is (571) 272-4436. The examiner can normally be reached on Mon-Fri from 9:30-6:00.

Art Unit: 2185

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Matthew Kim can be reached on (571) 272-4182. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

June 22, 2006


STEPHEN C. ELMORE
PRIMARY EXAMINER